

Design of an Ameliorated Methodology for the Abstraction of Usable Components of Object Oriented Paradigm from the Software Requirement Specification (SRS)

Dr. Shivanand M Handigund¹, Dr. R J D'Souza², Syed Naimatullah Hussain³

¹BIT/CSE, Bangalore, India

Email: smhandigund@gmail.com

²NITK/MACS, Surathkal, India

Email: rjd@nitk.ac.in

³NITK/MACS, Surathkal, India

Email: snuh@rediffmail.com

Abstract –The object oriented design requires that the view element required for the design is to be abstracted from the SRS. So it is required to transform the requirements into object oriented paradigm and then proceed for the development. We are intending in our ensued project, to develop a sequence of methods in the form of methodology, those take the requirements and then transform it into object-oriented paradigm. We are intending to develop an automated (with least human intervention) sequence of methodology that takes requirements specification as input and abstracts required elements for the object oriented system. This is a semiautomatic methodology. In few steps of our methodology whenever the human intervention is required the detailed guidelines for that of the process is framed to facilitate the human worker to take unique unambiguous decision.

I. INTRODUCTION

The software development project normally starts with customers' requirements. The customers are in general, strategic management people of the organization who are the user's of the ensuing tool. So the requirements of the ensuing system reflect their processing mindset. This will not serve the evolving process of organization. Presently, this will not serve the development process effectively. Now a day, people feel that the naturalness virtue of object-oriented paradigm made it more reliable durable and stable. The object oriented design requires that the view element required for the design is to be abstracted from the SRS. So it is required to transform the requirements into object-oriented paradigm and then proceed for the development. We are intending in our ensued project, to develop a sequence of methods in the form of methodology, those take the requirements and then transform it into object-oriented paradigm. We are intending to develop an automated (with least human intervention) sequence of methodology that takes requirements specification as input and abstracts required elements for the object oriented system. This is a semiautomatic methodology. In few steps of our methodology whenever the human intervention is required the detailed guidelines for that of the process is framed to

facilitate the human worker to take unique unambiguous decision.

Few researchers [1,6] have suggested some techniques for certain stages of the design of object classes. Although, these guidelines may facilitate to certain extent for the abstraction of object class name but since these methodologies are based on conjectures. There is not authentication of correctness and completeness of the end product of the ensued abstractions. We have made an attempt develop a methodology that identifies the object-oriented specifications in the form of object structures, object methods and the interrelationships, from the requirements of an information system. This semi automatic methodology comprises of a sequence steps like feasibility analysis, for object structure identification, resolution of synonyms & homonyms issues, regrouping of attributes of entities & functionalities through the design of data flow diagrams and elimination of imbalance between data & procedure selection along with authentication of correctness & completeness of the abstractions at each stage. This manual intervention at few stages is necessitated because of the need for human intelligence in these steps. Even for these manual intervention steps, attempt is made to provide clear-cut guidelines to streamline the design process. In the proposed methodology we have surmounted this lacuna and we have avoided conjecturing.

II. OBJECTIVE OF THE STUDY

Earlier the client is to make use of system analysis & design, the developer team is to study, analyze the system and design it. Now days the client organization are aware of information technology and its utility. Any project development starts with the client organization submitting the SRS. Client and developer organization come to certain agreement for terms and condition they prepare document called as project charter which contains all requirement of SRS, budgetary constraints, change management cost and the duration of the project. An attempt is made to develop and automatic methodology that takes SRS text as input

and abstract the elements required for object oriented design. Researcher's Rebecca Wirfs et al has developed an approach called noun phrase approach that identifies the object class names; and the process of identification of classes is an iterative process. They have not specified when the process needs to be stopped nor specified how the correctness and completeness is achieved and is based on conjecturing of human intelligence. Similarly few other researchers Shlaer, Mellor et al in common class pattern approach have stated that the entity as the object class structure. We know that the object class, the structure, the entity, the states and methods are encapsulated together, but in the entity concept functionality may utilize part of an entity attributes or it may contain number of attributes across different entities. Thus it fails to distinguish between entity and the object class structure. In the use case driven approach researchers Jacobson et al have stated that this approach identifies the information system as scenarios, the union of all scenarios becomes the behavior of the system. This is questionable concept as people are finding some activities of the system are beyond the scope of use-case scenarios. These methods even though make some sense; they lack the mathematical rigor and the naturalness. These are two essential components for the true object oriented design; the proposed methodology bridges the gap by incorporating mathematical rigor and naturalness. The authors have not noticed any single methodology that abstract the correct and complete object specifications. In our proposed methodology we are attempting to develop a correct and complete methodology that abstracts the requirement specification text in any paradigm in data/procedure oriented approaches and then automatically transforms the abstraction into first cut object specifications. Later these specifications are refined using the ambience of good database design principles. The abstracted specifications are correct and complete. Moreover we are attempting to develop a automatic methodology that has very limited human intervention thus avoiding the ambiguity. This intervention is made essential only for the authentication of the process for correctness and completeness. It minimizes the description of human intervention power.

III. LITERATURE SURVEY

A. Work Already Carried Out By Few Researcher's Noun Phrase Approach

Rebecca Wirf's Brock, Brian Wilkerson and Lauren Weiner (researcher's) have developed a eleven steps noun phrase approach for identification classes [2,8]. From the software requirement specification (SRS) or Use-Case nouns are identified as class and verbs as methods of classes. The process of identifying classes is an iterative process.

Short falls in noun phase approach

- Rebecca Wirf's et al have proposed an iterative method, which manually selects attribute name from the noun phrases and eliminates some of them based on some

reasoning. They have not specified when the iteration needs to be stopped, nor specified how the completeness and the correctness is achieved.

- They have identified object attributes from the noun phrases and object methods from verbs. This is questionable issue as there is vast flexibility in the English language for using verbs.

B. Common Class Pattern Approach

Short falls in common class pattern approach

- Common class pattern approach considers the entity as the object class structure. We know that the object class, the structure, the entity, the states and methods are encapsulated together, but in the entity concept functionality may utilize part of an entity attributes or it may contain number of attributes across different entities. Thus it fails to distinguish between entity and the object class structure.

C. Use-Case Driven Approach

Shortfalls in Use-Case Driven Approach

This identifies the information system as scenarios, the union of all scenarios becomes the behavior of the system. This is a questionable concept as people are finding some activities of the system are beyond the scopes of use-case scenarios.

These methods even though make some sense; they lack the mathematical rigour and the naturalness. These are the two essential components for the true object oriented design.

IV. PROPOSED SOLUTION METHOD/ALGORITHM

A. The Methodology

We have proposed a methodology, which will help in developing an automated methodology for the abstraction of object structure from the SRS. The input to the system is software requirements specification (SRS) (IV. B). The output of the methodology is the object structure functional dependencies, and also attribute domain pair. This methodology comprises a sequence of semiautomatic methods. Each stage of each method is proposed with details of either procedure, if it is automated, or guidelines, if it is a manual process.

B. System Requirement Specification (SRS)

Automation In Technical Institute / College

Functional Requirements:

Admit Student:

Student will get entrance exam card based on his/her performance, preference of college and availability of seats. The student will submit entrance exam card, original documents referenced in the card along with appropriate fees to the college and gets acknowledgement.

Register For Course

The academic section verifies the documents referenced in the card, if satisfied assign roll number to each student branch wise, year wise and after admission process is completed, sends the list to the concerned department.

Classroom Allocation

Each batch of each semester is assigned with a classroom thus a student is attached to a classroom in which the teaching learning process takes place as per the schedule

Faculty Work Load

A faculty member teaches the student the allocated subject in the allocated classroom at the allocated time day and hour.

C. Identify Different Nouns and Noun Phrases & Abstract Referenced and Defined Nouns From SRS (IV.B)

Identify noun/noun-phrases(N), Adjectives(Aj), Transitive Verbs(Vt), intransitive verb(Vi), Intransitive verb(Vi), impersonal verb(Vimp), Auxiliary Verb(Vaux), Adverb and Adverbial Phrases(Av), discard (Vaux), convert passive voice to active voice

TABLE 1
TABLE FOR NOUN PHRASE APPROACH

Noun	Adverb	Adjective	Vt	Vi	Vaux
Introduction	Formally	Admit	Provide	Is	Can
Purpose	suitably	Allocated	Intended	Receive	
Document	only	Eligible	Admitted	Fix	
Complete	apply	Submit			
Description	Strictly	performance			

To identify the functional dependencies among different attributes of object classes we are abstracting the referenced and defined nouns from the SRS(IV.B).

REFERENCED	DEFINED
Entrance exam card,	
performance, seat, college	→ Student
Entrance exam, fees, student	→
Acknowledgement, College	
Student	→ Documents, Roll no,
Semester,	
	Batch, Branch-
wise, Year-wise,	
	Branch, Schedule

D. Resolve Synonym, Homonym Issues From DOD (Data Oriented Data) & POF (Procedure Oriented Functionalities)

Step-1 Collect all attributes & entities their functional dependencies, interrelationship and procedure from customer team of user's CTU's.

Step-2 Design entities attributes along with primary keys and foreign keys. Develop ER model (DOD) collected from functionality (POF).

Step-3 Resolve synonym & homonym from both from DOD & POF.

Step-4 visualize all the functionalities that can be abstracted from the entity and attributes of DOD Similarly identify the attributes and entities required for the functionality POF

Establish On-to Correspondence Between DOD & POD (procedure oriented data)

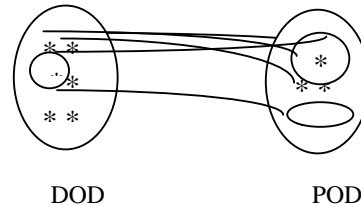


Figure 1

Match the attributes of DOD with attributes of POD & store them separately in DD (matched data) from DOD & POD (Procedure Oriented Data), for unmatched attributes see whether an attribute of one set has a matching part in the attribute of other set, decompose the other set of attributes and add matching attributes to corresponding pool.

Now we may have subsets containing unmatching attributes. Now consider in each set whether an attributes is synonymous with any attributes of other attributes, if so add to the pool of respective attribute, discard unmatched attributes of DOD and decompose the unmatched attributes of POD & add in both.

Establish on-to Correspondence Between POF & DOF (data oriented functionalities)

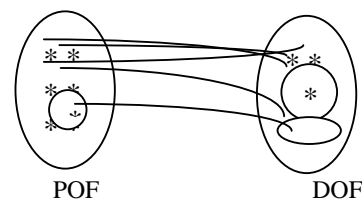


Figure 2

Match the functionality of POF (Procedure Oriented Functionality) with the functionality of DOF (Data Oriented Functionality) and store them separately in FF (matched Functionality), for unmatched functionalities compare for partial matching of POF& DOF. Decompose DOF and add in both, discard unmatched POF.

- Collect the functionality from CTU's; identify the data required for all these functionalities in terms of entity, attributes and interrelationship [1, 9].
- Group this attributes based on person, place, thing, event or concept and form entities attributes, primary key, foreign key; call this as POD

- Resolve synonym homonym issue amongst functionality name based on the behaviors of the functionalities. Now study each functionality for its relevance with respect to the information system.

Now consider first cut data set POD

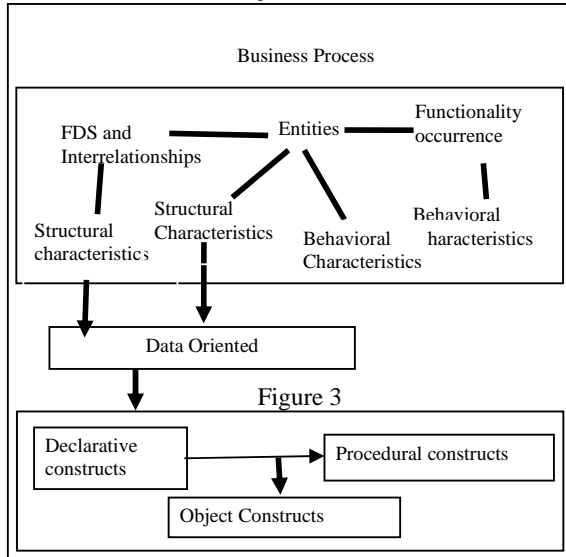


Figure 3

In the above fig 3, here more weightage is given to data, the upper rectangular block indicates the problem domain and the lower rectangular block states the solution domain. We identify the attributes using data oriented approach by making one-one and on-to functionality between the attributes of procedure oriented approach and attributes derived from data oriented approach.

Procedure oriented

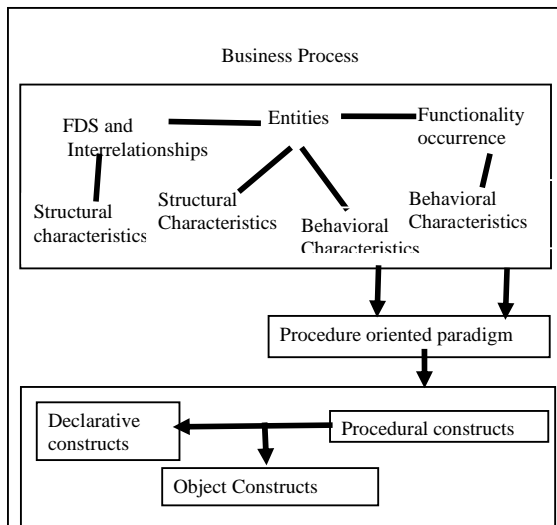


Figure 4

Fig 4, Now visualize all the data required for processing the functionalities POD (procedure oriented data) we

assume the file. We identify the functionalities, function oriented and procedure oriented. We identify the attributes using procedure oriented approach, identify the attributes required for the different functionalities, make one-one & on-to functionality between attributes of data oriented approach and attributes derived from procedure oriented approach [1, 3].

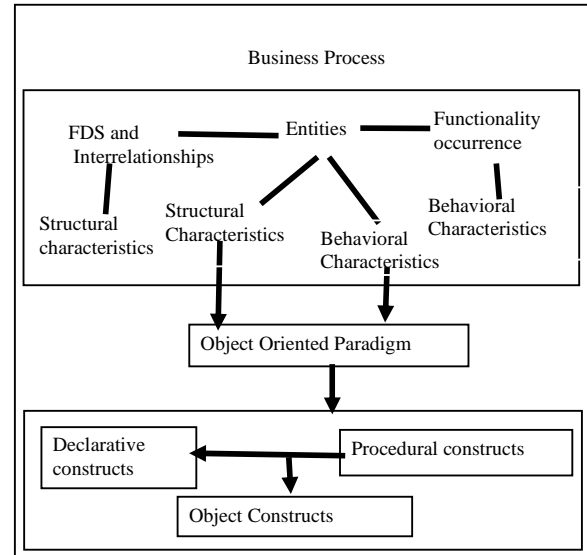


Figure 5

Object Oriented (fig 5) The above figure illustrates a perfect balance between the data and procedure oriented approach, here we identify the attributes using object oriented paradigms.

E. Group the Functional Dependencies for Identification of the Object Structure Attributes of Object Class from SRS(B)

Eg:

Functional Dependencies

Entrance Exam card,	
Performance, seats, college	→ student
Entrance exam, fees, student,	→ Acknowledgement
	College
Student	→ documents, assign roll no.,
	semester, batch, branch-wise,
	Year-wise, department, schedule
TLP	→ classroom
Classroom	→ batch, semester, student

Grouping the Attributes on RHS from FD'S

Student

Student, acknowledgement, college, semester,
Performance, borrowed books, shortage,
Lab exam, admission, staff

Performance

Performance, student

College

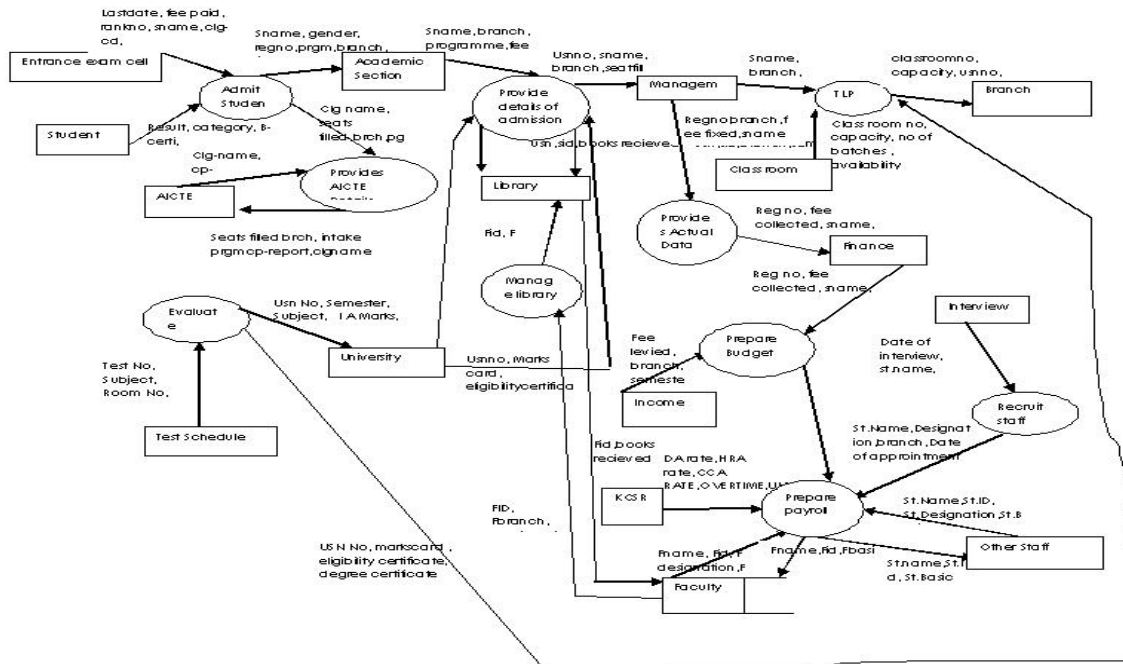
College, acknowledgement, order, engg degree

J. Normalization (SRS IV. B)

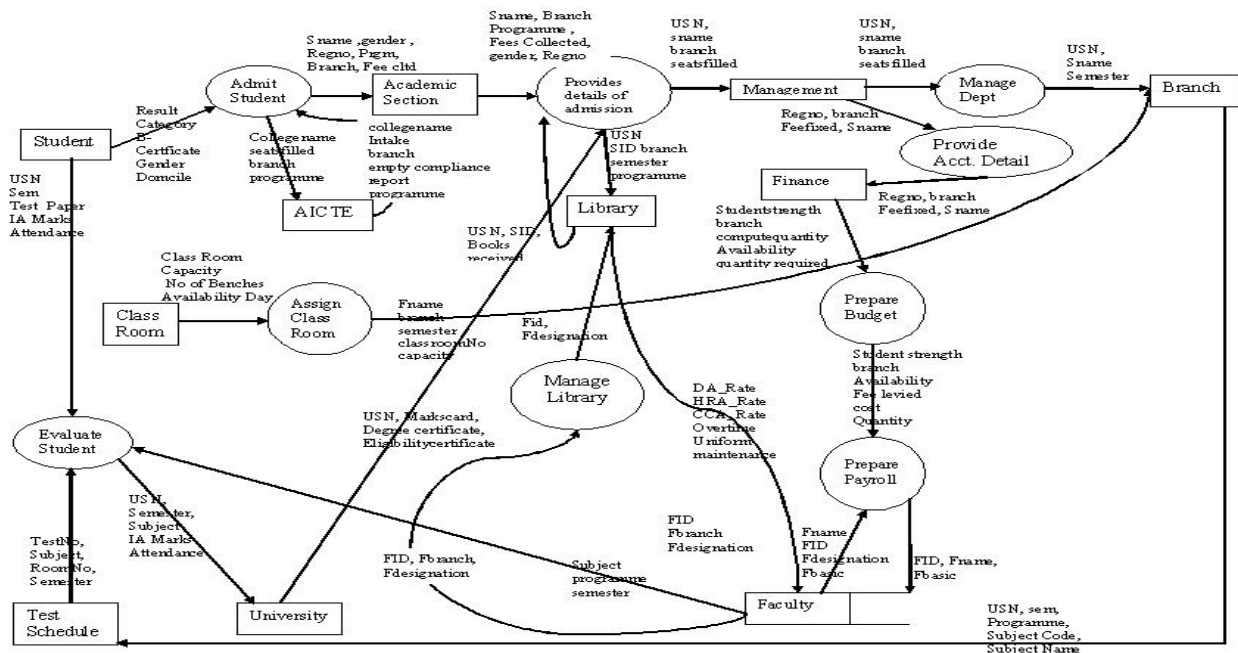
- College { college-name, branch, Programme, Intake, Compliance-report, Subject }
- Branch { ,Hod, branch-name, No.-of-faculty }
- Programme { Prgm-name, Prgm-co-ordinator, no. of student, qualification }
- Faculty {Fname, Fid, designation, brch-name, specialization, }
- Student {USno., sname, prgm,branch, qual }

- Salary { Fid, Fname, Fbasic, date-of-next-incrmnt.date-of-present-scale }
- Class Room{room-no, Fname,sub,fhour,pgm,brnch, sem avlbt day }
- Marks{USNo,sub-code,marks-obtn,grade-obtn,min-mrks,max-mrks }

K. Two Levels of Data Integration



L. Transform Logical DFD to Physical DFD:



V. CONCLUSION

This paper discusses the framework of our proposed research. The framework is developed on the study of different methodologies that exist for the abstraction of object classes from the software requirements specification. The available methodologies are designed with broken sequence of methods. Moreover piece-wise methods are manual methods without facilitating the opportunity for authenticating the correctness and completeness.

These methods can only be used to abstract object class names, currently no methods are available to abstract object attributes, methods moreover in the proposed methodology, we have used good database design principles to strengthen the high cohesion and low coupling properties, moreover we are planning to develop methodology for the abstraction of object method inculcating good software engineering design principles, thus the methodology transfers the business information into domain elements. Since we propose to use good data base design principles and good software engineering principles. The abstracted object classes will be free from the anomalies and blend of balanced approach. We are intending to refine the abstracted object classes to be more natural than random object classes.

In the proposed paper we have identified the functional dependencies (FDs) and minimization of FDs, and Two levels of data integration for refinement of object method. Still there is need to use good software engineering principles to identify object attributes involved in the object method

Our methodology is semi automatic it has advantages over existing methodologies in the sense that other methodologies have not given the clear cut methodology, sound reasoning for the steps. We have attempted to address these challenges in our methodology. As the SRS is prepared by number of users in organization, due to

flexibility in English language, different people use different words for the same meaning, automation is very difficult but however we propose an automated method in near future.

VI. REFERENCES

- 1) Shivanand M. Handigund, "Reverse Engineering of Legacy COBOL Systems", Ph.D. Thesis, 2001, I. IIT. Bambay, Mumbai.
- 2) Ali Bahrami "Object Oriented System Developments" McGraw-Hill International Editions 1999.
- 3) S.B. Navathe C. Batini, M. Lenzerini. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18(4):323-364,1986.
- 4) Joseph Fong et. al. Methodology of Scheme integration for new database applications: A practitioner's approach. *Journal of Database Management*, 10(1):3-18,1999.
- 5) Shivanand M. Handigund and Deepak B. Phatak. Reengineering methodology for legacy COBOL systems. In *Proceeding of International Conference on Information Technology & Communications Bangkok, Thailand* Pages 221-232, August 2000.
- 6) D B Phatak. Migrating Legacy systems. Pre-conference tutorial Presented at COMAD95, Pune (India), 1999.
- 7) Amith Sheth Vipul Kashyap. Semantic and schematic similarities between database Objects: A context based approach. *VLBD Journal*, 27(5):276-304,1996.
- 8) Rebecca Wirfs: Designing Object Oriented Software. Prentice-Hall India. 1990
- 9) Pankaj Jalote An Integrated Approach to Software engineering principles Pages 79-136, Springer's Publication,2005,India